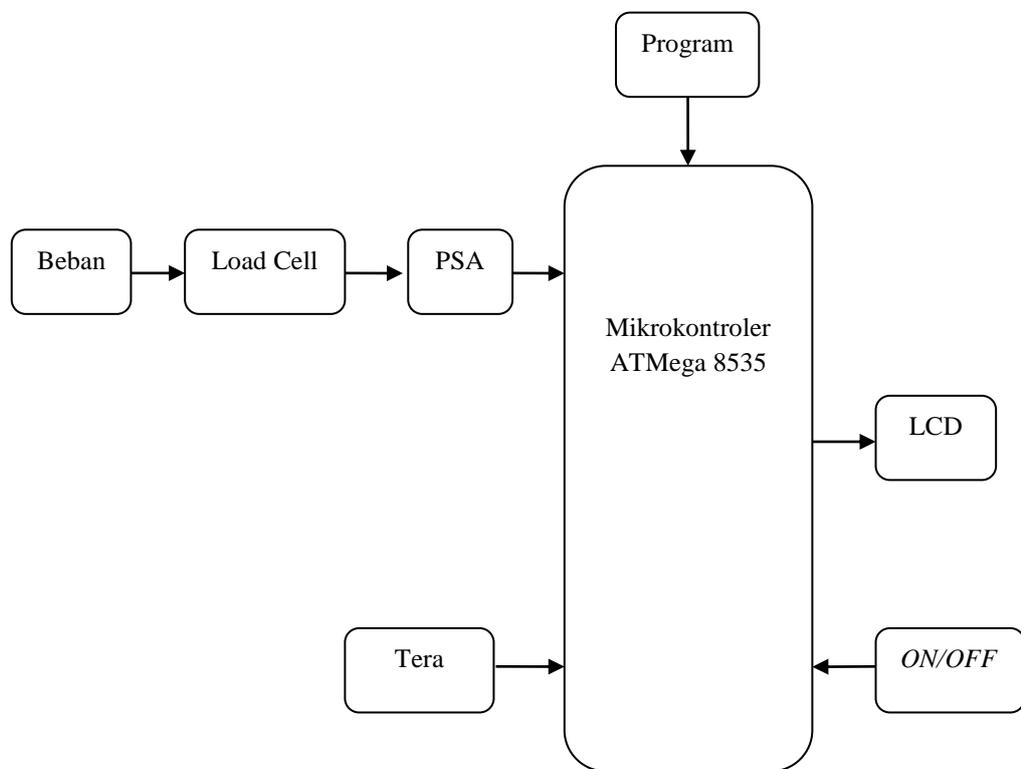


BAB III METODE PENELITIAN

3.1 Perancangan Perangkat Keras

3.1.1 Diagram Blok Sistem

Diagram blok sistem merupakan salah satu bagian terpenting dalam perancangan dan pembuatan alat ini, karena dari diagram blok dapat diketahui prinsip kerja keseluruhan rangkaian. Tujuan lain diagram blok ini adalah untuk memudahkan proses perancangan dan pembuatan pada masing-masing bagian, sehingga akan terbentuk suatu sistem yang sesuai dengan perancangan sebelumnya.



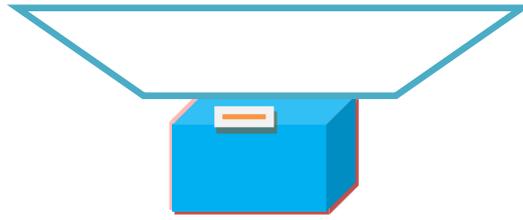
Gambar 3.1 Blok Diagram Timbangan Bayi

Cara Kerja Blok Diagram

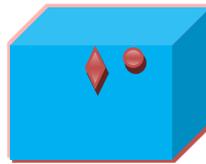
Pertama, menekan tombol *push ON* untuk menghidupkan *Power Bank*. Pada saat tombol *ON* ditekan maka *supply* (baterai) memberikan tegangan ke setiap rangkaian. Kemudian apabila terdeteksi ada beban yang mengenai sensor *loadcell* maka sensor mengalami perubahan resistansi. Selanjutnya resistansi akan dikonversi menjadi tegangan oleh jembatan *wheatstone* yang terdapat dalam sensor *loadcell* itu sendiri. Keluaran dari sensor *loadcell* masih sangat kecil yaitu dalam orde milivolt sehingga harus dikuatkan terlebih dahulu oleh *amplifier* yang terintegrasi dalam penguat instrumentasi (PSA). Kemudian sinyal analog akan diubah menjadi sinyal digital oleh *ADC* yang berada di dalam mikrokontroler *ATMega8535* dan data diolah sehingga dapat ditampilkan di *LCD* dalam besaran kilogram. Sedangkan fungsi dari tera adalah untuk me-nolkan hasil pada *display* apabila terdapat beban yang terdeteksi seperti berat selimut, *box* bayi dan lainnya sebelum dilakukan pengukuran berat bayi.

3.1.2 Rancang Bangun Timbangan Bayi

Bagaimana perancangan bentuk alat yang akan dibuat ditunjukkan pada gambar di bawah ini. Alat yang akan dibuat dalam bentuk balok yang didalamnya berisi rangkaian dan sensor, sedangkan di atas *box* terdapat tempat bayi yang dipasang tepat di atas sensor.



Gambar 3.3. Rancang Bangun Alat Tampak Depan



Gambar 3.4. Rancang bangun Alat Tampak Belakang

Keterangan :

	= <i>Display LCD</i>
	= Tombol Tera
	= Tombol <i>ON/OFF</i>

3.1.3 Modul Rangkaian *Minimum System*

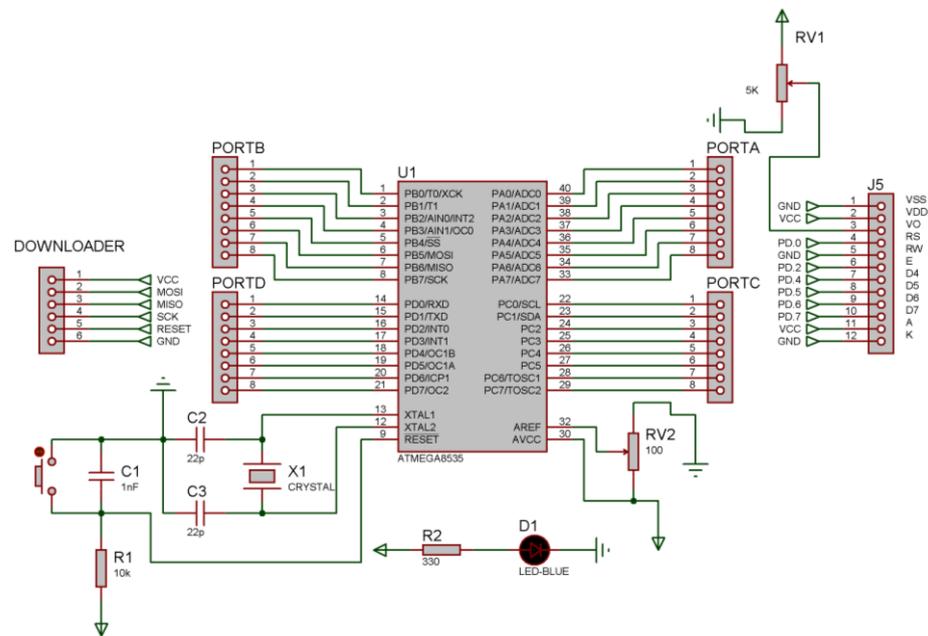
Rangkaian *minimum system* adalah rangkaian yang berfungsi sebagai kontrol atau otak dari alat yang dibuat. Mikrokontroler memiliki peran utama dalam sistem ini. Semua aktifitas sistem dikendalikan dengan program yang ada dalam mikrokontroler ini (Iswanto, Raharja & Subardono, 2009). Adapun spesifikasi yang diperlukan rangkaian ini diantaranya:

1. Membutuhkan tegangan 4,5 -5,5 VDC dan *ground*
2. Membutuhkan sambungan MISO, MOSI, SCK, dan RESET sebagai jalur *transfer data*
3. Membutuhkan tegangan pada pin Aref sebesar 3,45 volt
4. Membutuhkan *led* sebagai indikator tegangan

5. Membutuhkan *LCD* sebagai *display*

6. Membutuhkan *multiturn* sebagai pengatur kecerahan *LCD*

Jadi didapatkan rangkaian seperti Gambar 3.5 berikut ini:



Gambar 3.5. Schematik Minimum System ATmega8535

3.1.3.1 Alat yang digunakan

Berikut ini adalah alat yang dibutuhkan dalam pembuatan *Minimum System ATmega8535*

1. Tenol
2. Solder
3. Atraktor
4. Multimeter
5. Tang potong
6. Gerenda

7. Adaptor
8. Bor
9. Gergaji

3.1.3.2 Bahan yang digunakan

Berikut ini adalah alat yang dibutuhkan dalam pembuatan *Minimum System ATmega8535*

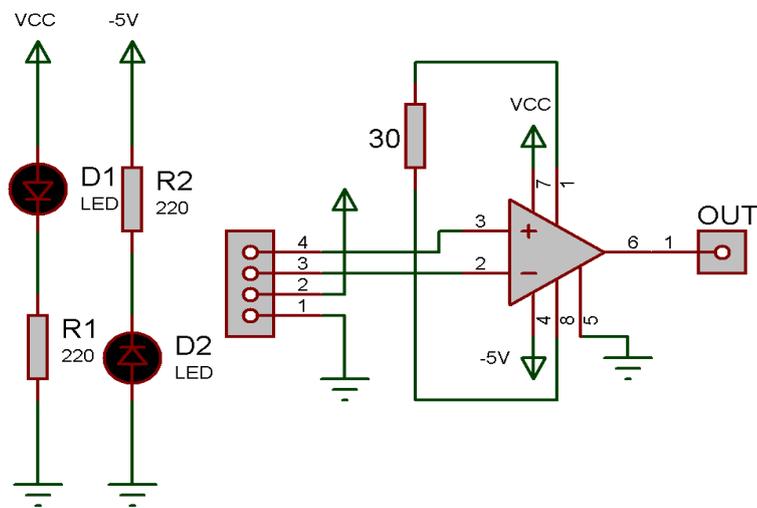
1. ATmega8535
2. *Socket IC*
3. Papan PCB Fiber
4. Pin sisir
5. *Push button*
6. *Capasitor 22pF dan 1uF*
7. *Resistor 330 Ω dan 10 K Ω*
8. *Led*
9. *Multiturn 10 K dan 5 K*
10. *Crystal 16 MHz*
11. *LCD Character 2x16*
12. *Jumper*

3.1.4 Modul Rangkaian Pengondisi Sinyal Analog (PSA)

Rangkaian Pengondisi sinyal merupakan rangkaian yang digunakan sebagai penguat dari *output* sensor *loadcell* dimana outputan dari sensor ini masih sangat kecil yaitu masih dalam orde *milivolt*. Adapun spesifikasi dari rangkaian ini diantaranya:

1. Sebuah sensor *loadcell* yang memiliki 4 kabel keluaran yaitu 2 kabel eksitasi dan 2 kabel *input*
2. Input tegangan 5 VDC, -5VDC dan *ground* untuk AD620
3. Input tegangan 5 VDC dan *ground* untuk sensor *loadcell*

Jadi didapatkan rangkaian seperti Gambar 3.6:



Gambar 3.6 *Schematik Pengondisi Sinyal Analog*

3.1.4.1 Alat yang digunakan

Berikut ini adalah alat yang dibutuhkan dalam pembuatan PSA (Pengondisi Sinyal Analog)

1. Solder
2. Tenol
3. Atraktor
4. Multimeter

5. Tang potong
6. Gerenda
7. Adaptor
8. Bor
9. Gergaji

3.1.4.2 Bahan yang digunakan

Berikut ini adalah alat yang dibutuhkan dalam pembuatan PSA (Pengondisi Sinyal Analog)

1. *IC AD620*
2. *Socket IC*
3. *Resistor 220 Ω dan 30 Ω*
4. *Led*
5. *Pin Sisir*
6. *Jumper*

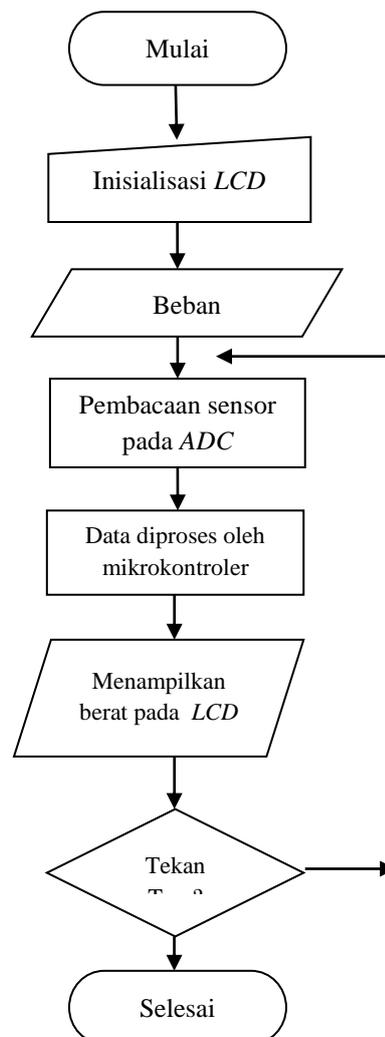
3.2 Perancangan Perangkat Lunak

3.2.1 Diagram Alir Sistem

Untuk mengetahui bagaimana urutan cara kerja alat, ditunjukkan dengan Diagram Alir Sistem dan cara kerja di bawah ini:

Ketika alat dinyalakan maka alat akan melakukan inisialisasi *LCD*. Apabila ada beban yang terdeteksi, maka sensor bekerja dengan mengeluarkan tegangan berdasarkan beban. Keluaran sensor akan dikuatkan

kemudian hasil penguatan tersebut dibaca oleh *ADC* pada *port* A.1 untuk diubah menjadi *signal* digital lalu diproses oleh mikrokontroler. Setelah diproses oleh mikrokontroler maka data berupa berat akan ditampilkan di *LCD*. Ketika tombol tera ditekan maka proses kembali ke pembacaan sensor pada *ADC*, sedangkan ketika tombol tera tidak ditekan maka proses selesai. Tombol tera disini ditekan oleh *user* apabila terdapat beban yang terukur sebelum beban bayi.



Gambar 3.7 Diagram Alir Sistem

3.2.2 Listing Program

1. Memanggil Library yang akan digunakan

```
#include <mega8535.h>
#include <delay.h>
#include <stdlib.h>
#include <stdio.h>
#include <alcd.h>
#include <math.h>
```

Tabel 3.1 Listing Program Library yang Digunakan

2. Melakukan inisialisasi data

```
float tera=0,data, tegangan, Kg, rata,
total=0;
unsigned char temp[6], temp2[6],temp3[5];
int i;
```

Tabel 3.2 Listing Program Penginisialisasian Data

3. Mengaktifkan ADC internal supaya mikrokontroler dapat bekerja dengan mengubah data analog menjadi data digital.

```
#define ADC_VREF_TYPE 0x00

// Read the AD conversion result
unsigned int read_adc(unsigned char
adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
// Delay needed for the stabilization of
the ADC input voltage
delay_us(10);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCW;
```

Tabel 3.3 Listing Program Pengaktifan ADC

4. Program pembacaan ADC dalam fungsi baca_Kg dengan mengubah data analog menjadi digital pada PORT A.1 dengan rumus $ADC = \frac{V_{in}}{V_{ref}} \times 1024$ dan

dilakukan looping sebanyak 300 kali untuk mencari rata-rata hasil supaya hasil yang terbaca stabil.

```
void baca_Kg()
{
    rata=0;
    for(i=0;i<300;i++)
    {
        data=read_adc(1);
        tegangan=(data*3.45/1024);
        rata=rata+tegangan;
    }
}
```

Tabel 3.4 *Listing Program Baca ADC*

5. Program untuk menampilkan berat dalam fungsi `baca_kg` menggunakan rumus persamaan garis yaitu

$$Y = mX + C$$

```
total=rata/300;
Kg=(5.775*total) - 4.43;
Kg=Kg+tera;
```

Tabel 3.5 *Listing Program Baca Kg*

6. Program untuk mengatur tata letak atau posisi tulisan pada tampilan *LCD*.

```
lcd_gotoxy(0,0);
lcd_putsf("BERAT =");
lcd_gotoxy(9,0);
ftoa(Kg,2,temp2);
lcd_puts(temp2);
lcd_gotoxy(14,0);
lcd_puts("Kg");

delay_ms(500);
```

Tabel 3.6 *Listing Program Tampilan LCD*

7. Program untuk menghitung tera, jadi dalam program ini dibuat ketika tombol tera ditekan maka berat akan menjadi 0 kg. Tombol diletakkan pada PORT B.0.

```

void hitung_tera()
{
  if (PINB.0==0)
  {
    tera=0;
    tera=tera-Kg;
    delay_ms(200);
  }
}

```

Tabel 3.7 *Listing Program Tombol Tera*

8. Program untuk memanggil setiap fungsi yang dibuat yaitu fungsi `baca_Kg` dan fungsi `hitung_tera` dengan melakukan looping secara terus menerus ketika logika benar.

```

while (1)
{
  baca_Kg();
  hitung_tera();
}

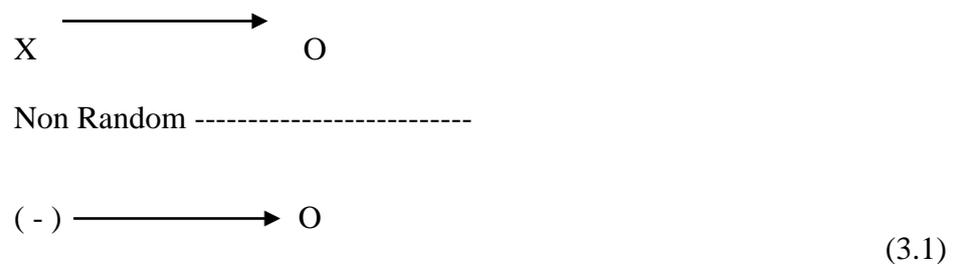
```

Tabel 3.8 *Listing Program Looping*

3.3 Jenis penelitian

Penelitian dan pembuatan alat ini dengan menggunakan design *pre eksperimental* dengan jenis penelitian adalah “*after only design*” karena perlakuan dibandingkan dengan alat pembanding anak timbangan.

Paradigma dalam penelitian eksperimen model ini dapat digambarkan sebagai berikut:



X = Treatment/perlakuan yang diberikan perlakuan berat (variabel Independen)

0 = Observasi dalam hal ini berupa sensor berat *loadcell* untuk mengukur berat (variabel dependen)

(-) = Kelompok *control* di sini menggunakan timbangan.

3.4 Variabel Penelitian

1. Variabel Bebas

Sebagai Variabel Bebas yaitu berat badan

2. Variabel Tergantung

Sebagai Variabel Tergantung yaitu sensor *Loadcell*

3. Variabel Terkendali

Variabel terkendali yaitu *IC* Mikrokontroler ATmega 8535.

3.5 Definisi Operasional Variabel

Dalam kegiatan operasionalnya, variabel-variabel digunakan dalam pembuatan modul, baik variabel terikat, tergantung, dan bebas memiliki fungsi antara lain:

Tabel 3.9 Definisi Operasional Variabel

Variabel	Definisi Operasional	Alat Ukur	Hasil Ukur	Skala-ukur
Berat Badan Bayi	Besaran fisis dari berat badan	Timbangan	0-15 kilogram	Rasio
Sensor <i>load cell</i> (variabel tergantung)	Pendeteksi berat badan bayi	Multimeter	0-100mVolt	Rasio

Variabel	Definisi Operasional	Alat Ukur	Hasil Ukur	Skala-ukur
Mikrokontroler	Komponen pengendali sistem yang harus di program untuk member perintah LCD untuk menampilkan hasil pengukura	Multimeter	0= Gnd 1=VCC	Nominal

3.6 Rumus Statistik

Setelah dilakukan pengukuran pada modul yang dibuat, dibutuhkan rumus statistik untuk membuktikan apakah modul yang telah dibuat layak digunakan atau tidak. Berikut rumus-rumus yang digunakan:

1. Rata – rata

Rata-rata adalah bilangan yang didapatkan dari hasil pembagian jumlah nilai data oleh banyaknya data dalam kumpulan tersebut. Rumus rata – rata adalah:

$$\text{Rata - rata } (X') = \frac{X_1 + X_2 + \dots + X_n}{n} \quad (3.2)$$

Dimana:

X' = Rata-rata

X_1, \dots, X_n = Nilai data

N = Banyak data

2. Standar Deviasi

Adalah suatu nilai yang menunjukkan tingkat (derajat) variasi kelompok data atau ukuran standard penyimpangan dari rata-ratanya. Jika standard deviasi semakin kecil maka data tersebut semakin presisi.

Rumus Standart Deviasi adalah :

$$\text{Standart Deviasi (SD)} = \sqrt{\frac{(X_1 - X')^2 + (X_2 - X')^2 + \dots + (X_n - X')^2}{n - 1}} \quad (3.3)$$

Dimana:

SD = Standar deviasi

X' = Rata-rata

X₁, ..., X_n = Nilai data

N = Banyak data

3. Simpangan

Simpangan adalah selisih dari rata-rata nilai harga yang dikehendaki dengan nilai yang diukur. Berikut rumus dari simpangan:

$$\text{Simpangan} = X - X' \quad (3.4)$$

Dimana :

X = Data

X' = Rata-rata

4. Nilai *Error*

Error (Rata-rata Simpangan) adalah selisih antara mean terhadap masing – masing data. Rumus *Error* adalah :

$$\boxed{\% \text{ERROR} = \frac{X - X'}{X} \times 100\%} \quad (3.5)$$

Dimana:

X = Data

X' = Rata-rata